SYSTEM.INI File Settings

Windows initialization files contain information that defines your Windows environment. Windows and Windows applications can use the information stored in these files to configure themselves to meet your needs and preferences.

There are two standard Windows initialization files:

- WIN.INI, which primarily contains settings that Windows maintains to customize your Windows environment according to your preferences.

- SYSTEM.INI, which primarily contains settings that customize Windows to meet your system's hardware needs.

This document describes only a few of the settings in the SYSTEM.INI file. You can get a more complete listing of the SYSTEM.INI settings by purchasing the *Windows Resource Kit for the Microsoft Windows Operating System Version 3.1.* The Resource Kit provides complete technical information about Windows version 3.1 for the support professional. It includes a technical reference manual and a disk containing helpful utilities, system resource viewers, drivers, and accessories.

To order the Windows Resource Kit within the United States, dial:

1-800-642-7676

To order outside of the United States, dial the phone number for your area. You can find this number on the International Subsidiary card.

Format of the SYSTEM.INI File

The SYSTEM.INI file contains several sections, each of which consists of a group of related settings. The sections and settings are listed in the SYSTEM.INI file in the following format:

```
[section name]
keyname=value
```

In this example, **[section name]** is the name of a section. The enclosing brackets ([ ]) are required, and the left bracket must be in the leftmost column on the screen.

The **keyname=value** statement defines the value of each setting. A keyname is the name of a setting. It can consist of any combination of letters and digits in uppercase or lowercase, and it must be followed immediately by an equal sign (=). The value of the setting can be an integer, a Boolean value, a string, or a quoted string, depending on the setting. There are multiple settings in most sections.

You can include comments in initialization files. You must begin each line of a comment with a semicolon (;).

How to Read the Listings in This Document

In this document, settings in SYSTEM.INI are listed alphabetically within sections. However, the settings do not appear alphabetically in the SYSTEM.INI file. To change a setting, you need to search for it in the appropriate section.

The syntax, default value, and purpose appear in the following format.

SettingName=<value-type>

Default: The Windows default value for this setting.
Purpose: A brief description of the function of the setting.

<value-type> indicates whether the value should be an integer, a letter, a range of numbers, a Boolean value, or something else. If you want to enable a Boolean setting, you can set the value to: True, Yes, On, or 1. If you want to disable the Boolean setting, you can set the value to: False, No, Off, or 0.

Changing Settings

When the Setup program creates the SYSTEM.INI file, it assigns default values to many of the settings. Other settings have values that correspond to the type of hardware you are using.

Some settings listed in this document do not normally appear in your SYSTEM.INI file. Most of these settings have a built-in default value that is present whether or not the setting appears in SYSTEM.INI. You might need to change one or more of these values to improve the performance of Windows or applications with your system.

To change the values of the settings described in this document, open the SYSTEM.INI file and edit it by using a text editor, such as Windows Notepad. Do not use a formatting editor, such as a word processor in document mode; it can corrupt your SYSTEM.INI file. For more information on how to use Notepad, see the *Microsoft Windows User's Guide.*

Any changes you make to SYSTEM.INI settings do not take effect until you restart Windows.

Caution: Do not change any settings that are not described in this document.

If you need to change additional settings, refer to the *Windows Resource Kit for the Microsoft Windows Operating System Version 3.1*. Always back up your SYSTEM.INI file before you change any of its settings. That way, you can restore the original file if you make incorrect changes. Be careful when making changes to the SYSTEM.INI file. Your system will not work properly if you edit this file incorrectly.

[boot] Section Settings

All settings in this section, except for CachedFileHandles, are required. If you modify or delete one of these settings, Windows might not work properly. There are no specific default values for these settings; Setup assigns values based on your system configuration.

If you want to run a different application when you start Windows, change the following setting.

---

shell=<filename>

Default: None (Setup initializes this value as PROGMAN.EXE.)

Purpose: Specifies the Windows application that runs when you start Windows. Unless you change this setting, Program Manager will run when you start Windows. For example, if you want to run File Manager instead of Program Manager when you start Windows, you could change this setting as follows:

shell=winfile.exe

[NonWindowsApp] Section Settings

The [NonWindowsApp] section contains settings that affect the performance of non-Windows applications.

The [NonWindowsApp] section can contain the following settings.

CommandEnvSize=<bytes>

Default: 0 for MS-DOS versions earlier than 3.2. Otherwise, the default is the value for the **/e:** option in the **shell=** command line in the CONFIG.SYS file.

Purpose: Specifies the size of the COMMAND.COM environment. Note that running batch files with the extension .BAT starts COMMAND.COM, so this setting also applies to batch files. The value for this setting must be either 0 or between 160 and 32768. A value of 0 disables this setting. If the value is too small or too large, it will be rounded up to 160 or down to 32768. If the value is less than the current size of the actual environment, this setting will be disabled, as if it were set to 0. If you have specified the environment size in a PIF file for COMMAND.COM, the PIF-file setting overrides this setting.

DisablePositionSave=<Boolean>

Default: 0

Purpose: When this setting is disabled, the position and fonts used by a non-Windows application are saved in the DOSAPP.INI file when you quit the application. If this setting is enabled, the position and fonts used by a non-Windows application whose settings have not been previously saved in the DOSAPP.INI file will not be saved. If enabled, the setting can be overridden for each non-Windows application by selecting the Save Settings On Exit check box in the Font dialog box in the application.

FontChangeEnable=<Boolean>

Default: 1 on systems that use Windows version 3.1 grabbers. 0 on systems that use Windows version 3.0 grabbers.

Purpose: Provides the ability to change fonts when running a non-Windows application in a window on a system that uses version 3.0 of the grabbers (usually 3.0 display drivers). Windows version 3.1 video grabbers (used in 3.1 display drivers) include built-in support for changing fonts when running a non-Windows application in a window. If you are using a 3.0 grabber that has not been updated to include the ability to change fonts and you want to use this feature, enable this setting. However, with this setting enabled, your screen may lose characters and the cursor may change size and position slightly.

LocalTSRs=<list-of-TSR-applications>

Default: DOSEDIT, CED

Purpose: Specifies which terminate-and-stay-resident (TSR) programs work properly if they are copied to each instance of a virtual machine. When you start Windows, it detects any TSR programs that are currently running. If the TSR is on the LocalTSRs list, Windows copies the TSR to each virtual machine you run. Many TSRs will not run properly if they are added to this list. Make sure your TSR is fully compatible with Windows and can be copied to a virtual machine before adding it to the list.

MouseInDosBox=<Boolean>

Default: 1 if an MS-DOS mouse driver is loaded that has the extension .COM or .SYS and supports the use of a mouse with a non-Windows application. Otherwise, the default is 0.

Purpose: Specifies whether the mouse is supported when running a non-Windows application in a window. Mouse support for a non-Windows application running in a window is automatically available if you are using a Windows 3.1 version of the grabbers. If you are using a Windows 3.0 version of the grabbers and you want mouse support, enable this setting. If you do not want mouse support, disable this setting.

NetAsynchSwitching=<0-or-1>

Default: 0, unless an application is running that supports the use of the Task Switcher API by the NetBIOS.

Purpose: Specifies whether you can switch away from an application (running in standard mode) after it has made an asynchronous network BIOS call. If this setting is disabled, you cannot switch away. Switching away from some applications that make these calls might cause your system to fail. Once Windows detects an asynchronous NetBIOS call, you cannot switch away from the application, even if no more of these calls are made. Enable this setting only if you are sure that the applications you use will not receive network messages while they are inactive.

ScreenLines=<number>

Default: 25

Purpose: Specifies the number of lines that will be displayed on the screen when you run a non-Windows application. An application that specifies a different screen mode can override this setting.

SwapDisk=<drive-colon-directory>

Default: The directory that the TEMP environment variable points to; if there is no TEMP variable, the default is the root directory of your first hard drive (usually labeled C:). If you do not have a hard disk, the default is the root directory of your first floppy disk drive (usually labeled A:).

Purpose: Provides the name of the disk drive and directory to which Windows running in standard mode swaps non-Windows applications.

[standard] Section Settings

The [standard] section contains settings that are specific to running Windows in standard mode.

The [standard] section can contain the following settings.

FasterModeSwitch=<0-or-1>

Default: 0

Purpose: Enabling this setting causes Windows running in standard mode to use a faster method of switching from protected to real mode on many 80286-based computers. When this setting is enabled, Windows responds quicker to hardware interrupts, allowing better throughput for interrupt-intensive applications, such as communications applications. In addition, you should enable this setting if you are using a Zenith Z-248 system and are losing characters while typing, or if you are using an Olivetti M-250-E and lose mouse functionality.

Note: This setting has no effect on 80386-based computers. Some early IBM AT and compatible computers do not have the BIOS support necessary to use this setting. Enabling this setting on these computers may cause them to lock up when starting Windows.

Int28Filter=<number>

Default: 10

Purpose: Specifies the interval of INT28h interrupts, generated when the system is idle, that are made visible (or *reflected*) to software that is loaded before Windows. Windows will reflect every *n*th interrupt, where *n* is the value of this setting. For example, a value of 1 reflects every INT28h interrupt, a value of 2 reflects every second INT28h interrupt, and so on. Increasing this value might improve Windows performance, but may interfere with some memory-resident programs, such as network software. Set this value to 0 to prevent any INT28h interrupts from being reflected. Setting this value too low (from 1 to 9) might interfere with communications applications.

NetHeapSize=<kilobytes>

Default: 8

Purpose: Specifies the size (in kilobytes) of the data-transfer buffer that Windows running in standard mode allocates in conventional memory for transferring data over a network. If an application is not running correctly, your network may require a larger buffer than the default value. Increasing this value will decrease the amount of memory available to applications. If no network software is running, this setting will be ignored and no memory will be allocated.

[386Enh] Section Settings

The [386Enh] section contains information specific to running Windows in 386 enhanced mode, including information used for virtual-memory page swapping.

The [386Enh] section can contain the following settings.

AllVMsExclusive=<Boolean>

Default: False

Purpose: If enabled, this setting forces all applications to run in exclusive full-screen mode, overriding all contrary settings in the applications' program information files (PIFs). Enabling this setting might prolong the length of the Windows session when you are running network and memory-resident software that is incompatible with Windows.

COMBoostTime=<milliseconds>

Default: 2

Purpose: Specifies the amount of time (in milliseconds) to allow a virtual machine to process a COM interrupt. If, while running a communications application, you lose keyboard characters on the screen, you can try increasing this value.

COMdrv30=<Boolean>

Default: False

Purpose: If enabled, the Virtual COM Driver (VCD) uses its own copy of the interrupt handler for the serial communications driver. This improves performance of your COM ports. Enable this setting if you are using a Windows version 3.0 serial communications driver. Disable this setting if you are using the standard Windows 3.1 serial communications driver.

COM1FIFO=<Boolean>
COM2FIFO=<Boolean>
COM3FIFO=<Boolean>
COM4FIFO=<Boolean>

Default: True

Purpose: Specifies whether the FIFO buffer of a COM port's 16550 Universal Asynchronous Receiver Transmitter (UART) should be enabled (True) or disabled (False). If a serial port does not have a 16550 UART, this setting is ignored.

Note: These values are used by Windows for both standard and

386 enhanced modes.

COMIrqSharing=<Boolean>

Default: True for Micro Channel and EISA machines; False for all other machines.

Purpose: Specifies whether COM interrupt lines are sharable between multiple serial ports or with other devices. Enable this setting if your machine uses the same interrupt for COM3 or COM4 as it does for COM1 or COM2.

DOSPromptExitInstruc=<Boolean>

Default: Yes

Purpose: If this setting is enabled, when you start the MS-DOS prompt a message box appears with instructions on how to exit and switch away from the MS-DOS prompt. Disable this setting if you do not want to see the message.

DualDisplay=<Boolean>

Default: See "Purpose" below.

Purpose: Normally, when running in 386 enhanced mode, the memory between B000:0000 and B7FF:000F is used by the general system unless a secondary display is detected. Enable this setting if you are using a VGA-based color display and want EMM386.EXE to include this address space as an upper memory block (UMB). In addition to enabling this setting, you must include the **i=** option in the **device=EMM386.EXE** command line in your CONFIG.SYS file as follows:

device=EMM386.EXE i=B000-B7FF

If this setting is disabled, the address range is available on EGA systems, but not on VGA systems, because the VGA display device supports monochrome modes, which use this address space.

EMMExclude=<paragraph-range>

Default: None

Purpose: Specifies a range of memory that Windows will not scan to find unused address space. This has the side effect of turning off the RAM and ROM search code for the range. The range (two paragraph values separated by a hyphen) must be between A000 and EFFF. This scanning can interfere with some adapters that use the same memory area. The starting value is rounded down and the ending value is rounded up to a multiple of 16K. For example, you could set EMMExclude=C800-CFFF to prevent Windows from scanning the addresses C800:0000 through

CFFF:000F. You can specify more than one range by including more than one EMMExclude line.

EMMInclude=<paragraph-range>

Default: None

Purpose: Specifies a range of memory that Windows will treat as unused address space regardless of what may be there. EMMInclude takes precedence over EMMExclude if you specify ranges that overlap. The range (two values separated by a hyphen) must be between A000 and EFFF. The starting value is rounded down and the ending value is rounded up to a multiple of 16K. For example, you could set EMMInclude=C800-CFFF to ensure that Windows can use the addresses C800:0000 through CFFF:000F. You may specify more than one range by including more than one EMMInclude line.

EMMPageFrame=<paragraph>

Default: None

Purpose: Specifies the starting paragraph where the 64K page frame will begin when Windows (running in 386 enhanced mode) cannot find a suitable page frame. Allows an EMM page frame in an area containing some unused RAM or ROM. For example, you could set EMMPageFrame=C400 to start the page frame at C400:0000.

EMMSize=<kilobytes>

Default: 65536

Purpose: Specifies the total amount of memory available for mapping as expanded memory. The default value allocates the maximum possible amount of system memory as expanded memory. Specify a value for this setting if you run an application that allocates all of the available expanded memory. If this is the case, you cannot create new virtual machines. If this value is zero, no expanded memory is allocated, but the EMM driver will load. To disable EMM and prevent the EMM driver from loading, use the NoEMMDriver setting.

FileSysChange=<Boolean>

Default: Off in 386 enhanced mode; not supported in standard mode.

Purpose: Indicates whether File Manager automatically receives messages anytime a non-Windows application creates, renames, or deletes a file. If this setting is disabled, a virtual machine can run exclusively, even if it modifies files. Enabling this setting can slow down system performance significantly.

InDOSPolling=<Boolean>

Default: No

Purpose: If enabled, prevents Windows from running other applications when memory-resident software has the InDOS flag set. Enabling this setting is necessary if the memory-resident software needs to be in a critical section to do operations off an INT21 hook, but will slow down system performance slightly.

INT28Critical=<Boolean>

Default: True

Purpose: Specifies whether a critical section is needed to handle INT28h interrupts used by memory-resident software. Some networks do internal task switching on INT28h interrupts. These interrupts might lock up some network software, indicating the need for an INT28h critical section. If you are not using such software, you might improve Windows task switching by disabling this setting.

LocalReboot=<on-or-off>

Default: On

Purpose: Specifies whether you can press CTRL+ALT+DEL to quit applications that cause an unrecoverable error in 386 enhanced mode. If this setting is enabled, you can quit the applications without restarting Windows. If this setting is disabled, pressing CTRL+ALT+DEL will restart your entire system.

MaxCOMPort=<number>

Default: 4

Purpose: Specifies the maximum number of COM ports supported in 386 enhanced mode. Change this value if you have more than four COM ports installed in your computer.

NetAsynchFallback=<Boolean>

Default: False

Purpose: If this setting is enabled, Windows attempts to save a failing NetBIOS request. When an application issues an asynchronous NetBIOS request, Windows attempts to allocate space in its global network buffer to receive the data. If there is insufficient space in the global buffer, Windows typically fails the NetBIOS request. If this setting is enabled, Windows attempts to save such a request by allocating a buffer in local memory and preventing any other virtual machines from running until the data is received or the timeout period (specified by the NetAsynchTimeout setting) expires.

NetAsynchTimeout=<seconds>

Default: 5.0
Purpose: Specifies the timeout period (in seconds) when Windows will enter a critical section in order to service an asynchronous NetBIOS request. It is used only when the **NetAsynchFallback** setting is enabled. This value can include a decimal (such as 0.5).

NetDMASize=<kilobytes>

Default: 32 on Micro Channel machines (IBM PS/2 or compatible); 0 on non-Micro Channel machines (IBM PC/AT or compatible).
Purpose: Specifies the DMA buffer size (in kilobytes) for NetBIOS transport software if a network has been installed. In this case, the buffer size is the larger of this value or the value of DMABufferSize.

NetHeapSize=<kilobytes>

Default: 12
Purpose: Specifies the size (in kilobytes) of the data-transfer buffers in conventional memory that Windows allocates for transferring data over a network when running in 386 enhanced mode. All values are rounded up to the nearest 4K.

Network=<filename-or-devicename>

Default: None (Setup assigns an appropriate value based on your system configuration.)
Purpose: Specifies the type of network you are using with Windows in 386 enhanced mode.

PSPIncrement=<number>

Default: 2
Purpose: Specifies the amount of additional memory, in 16-byte increments, that Windows should reserve in each successive virtual machine when the UniqueDOSPSP setting is enabled. The setting that works best for your machine might vary depending on your memory configuration and the applications you are running. Valid values are 2 through 64. See the UniqueDosPSP setting for more information.

ReflectDosInt2A=<Boolean>

Default: False
Purpose: Indicates whether Windows should consume or reflect DOS

INT 2A signals. The default means Windows will consume these signals and therefore run more efficiently. Enable this setting if you are running memory-resident software that relies on detecting INT2A messages.

SyncTime=<Boolean>

Default: True

Purpose: If this setting is enabled, Windows periodically synchronizes its time with the computer's CMOS clock. If this setting is disabled, Windows usually maintains the correct time, unless TrapTimerPorts is disabled and you are running applications that can cause the system time to run faster or slower than the actual time. This setting is related to the TrapTimerPorts setting.

TimerCriticalSection=<milliseconds>

Default: 0

Purpose: Instructs Windows to go into a critical section around all timer interrupt code and specifies a timeout period (in milliseconds). Specifying a positive value causes only one virtual machine at a time to receive timer interrupts. Some networks and other global memory-resident software may fail unless this setting is used. However, using this setting slows down performance and can make the system seem to stop for short periods of time.

TrapTimerPorts=<Boolean>

Default: True

Purpose: Specifies whether Windows should trap read and write operations to the system timer ports that are performed by applications. If this setting is disabled, Windows will not trap these operations, allowing applications that frequently read or write to the timer to run faster. However, this may interfere with ability of Windows to keep accurate system time. If this setting is disabled, Windows can usually detect when an application has changed the timer interrupt interval and then make any adjustments to the time. If your system's time appears to be running fast or slow, enable this setting. If you do not want to enable this setting, enable the SyncTime setting. This causes Windows to check the time periodically and then make any necessary adjustments.

UniqueDOSPSP=<Boolean>

Default: True if you are running a network based on Microsoft Network or LAN Manager. See the NETWORKS.WRI online document to find out whether the network you are running is one of these. False for all other networks.

Purpose: If this setting is enabled, Windows starts every application at a unique address (PSP). Each time Windows creates a new virtual machine to start a new application, Windows reserves a unique amount of memory (i bytes) below the application. For example, if the first application is loaded at address M, the second application is loaded at address M+i, the third at M+2i, and so on. The amount of memory (i) is determined by the PSPIncrement setting. These settings ensure that applications in different virtual machines all start at different addresses. Some networks use applications' load addresses to identify the different processes that are using the network. Disabling this setting on such networks might cause one application to fail when you quit another, because the network interprets them as the same application. When this setting is enabled, slightly less memory is available for non-Windows applications.

VirtualHDIrq=<Boolean>

Default: On for AT-compatible computers; Off for all other computers.

Purpose: If enabled, Windows in 386 enhanced mode can terminate interrupts from the hard disk controller, bypassing the ROM routine that handles these interrupts. Some hard disk drives might require this setting to be disabled in order for interrupts to be processed correctly. If this setting is disabled, the ROM routine handles the interrupts, which slows down system performance.